Remarks

Claims 1-10 stand rejected under 35 U.S.C. §§ 102 and 103 as being either anticipated by or unpatentable over U.S. Patent 6,357,005 to Devaux et al. (Devaux), alone or in combination with other cited references (paper no. 5, ¶¶ 2 and 4-6). Applicants respectfully traverse these rejections.

In applicants' claimed method for managing a dynamic file system (12), there are embedded in the dynamic file system one or more static data objects (22) that are excluded from actions performed dynamically on the file system. By thus "playing around" certain static data objects, applicants are able to enjoy the advantages of a dynamic file system (in terms of flexible memory allocation and the like), while retaining the ability to statically access certain objects (as at boot time, when the file system has not yet been loaded).

Devaux discloses a system for the secure CD-ROM storage of data in which an electronic decryption microcircuit 13 is embedded in the plastic of the central region 12 of a CD-ROM 1 (Fig. 1). A chip card 3 that is inserted into a connector 23 of a CD-ROM drive 2 contains at least part of a cryptographic key, while any remaining part of the key is stored in the microcircuit 13. The CD-ROM drive 2, the chip card 3 and the microcircuit 13 interact with one another to decrypt data stored in encrypted form on the CD-ROM 1.

From this summary, it will be seen that about the only thing applicants' claimed invention and Devaux's system have in common is the concept of embedding. What they embed, however, is completely different. In applicants' claimed method, static objects are embedded in a dynamic file system.[1] Devaux, on the other hand, embeds a physical device (microcircuit 13) in a physical substrate (CD-ROM 1), and in the central region 12 rather than the annular region 10 where the data is stored. Devaux gives no information at all about the organization of the data stored on the CD-ROM 1, much less any suggestion of a file system. For all that the patent says about the

[1] As defined in the online encyclopedia Webopedia (http://www.webopedia.com/), a "file system" is "[t]he system that an operating system or program uses to organize and keep track of files" (hyperlinks omitted) A "file system" is thus a programming construct and not simply anything (hardware, software or whatever) that might contain a file.

matter, CD-ROM 1 may simply store an undifferentiated set of bytes that are read sequentially from start to finish.

Even if there were a file system somewhere (e.g., in the operating system of a computer to which the CD-ROM player is attached), it would not be a dynamic file system, since a CD-ROM is by definition read only and the stored information never changes. Further, neither microcircuit 13 nor the data stored in the microcircuit would be "embedded" in that file system. There is no chance, for example, that in the absence of some special exclusion mechanism, a person reading data off CD-ROM 1 would read the key stored in microcircuit 13. Not only is the storage area of microcircuit 13 not contiguous with that of CD-ROM 1, but the technologies used are distinct. Microcircuit 13 thus communicates with the CD-ROM drive 2 by means of an inductive antenna 14, which is entirely separate from the laser or LED that is used to read the CD-ROM itself.

In sum, Devaux teaches embedding an electronic circuit in the substrate of an optical storage medium. It does not teach embedding a static object in a dynamic file system as claimed by applicants, nor would the disclosed system be capable of functioning like the one claimed by applicants.

Neither of the other cited references, Benson et al. U.S. Patent 5,819,252 (Benson) and Cepulis et al. U.S. application 2001/0042225 (Cepulis), remedies these deficiencies of Devaux as a reference. As recited in its abstract, Benson describes a method for detecting and handling an invalid use of a data structure. A data structure 30 (Fig. 2A) associated with a first (e.g., 64-bit) computing environment includes a field 30d having a stored value identifying an inaccessible address in a second (e.g., 32-bit) computing environment. The field 30d is used to detect an invalid use of the data structure in the second computing environment by a computer program 21 attempting to access memory using the inaccessible address indicated by the value contained in the field 30d.

The Examiner urges that Benson discloses "defining an embedded static object by a memory address and a fixed size", as recited in applicants' claim 2. Nothing of this type is discernible from the patent. Assuming, for the sake of argument, that Benson uses a file system (or perhaps

separate 32-bit and 64-bit file systems), there is nothing to suggest any dichotomy between static objects and a dynamic file system as claimed by applicants. If Benson prohibits certain memory accesses, it is because of incompatibilities between programs and resources, not because of any attempt to access a "static" object in a dynamic file system.

Cepulis is similarly inapposite. As recited in claim 25 of the published application, a computer system 100 (Fig. 1) comprises a plurality of CPUs 102 including a boot-strap processor (BSP), a plurality 118 of memory modules 112, a North bridge device 106 coupling together the CPUs 102 and the memory modules 112, a master control device 140 coupled to the North bridge device via a primary expansion bus 108 and coupled to the CPUs 102 and the memory modules 112 via a serial bus (I^2C). Each CPU 102 and each memory module 112 include non-volatile memory 104 for storing an ID code uniquely identifying that CPU or memory module. A non-volatile memory device 130 coupled to the North bridge device 106 stores a failed device log 132 that includes a list of ID codes associated with a CPU 102 or memory module 112 that has failed. The master control device 140 retrieves the ID codes during initialization and provides the ID codes to the boot-strap processor BSP, which compares the retrieved ID codes with the list of ID codes in the failed device log 132 to determine if a CPU 102 or memory module 112 that is listed in the failed device log as failed is present in the computer system 100.

The Examiner urges that Cepulis teaches "accessing a static object in a pre-boot phase of a host system connected to the chip card" as recited in applicants' claims 6 and 7. This is simply not correct. While the master control device 140 and boot-strap processor BSP of Cepulis may access ID codes during initialization, these "objects" are stored in a non-volatile memory device 130 and are not "embedded" in a dynamic file system as claimed by applicants. No special action is thus necessary to "exclude" such objects from actions performed on such a file system.

Conclusion

In summary, not only does Devaux fail to teach applicants' claimed invention, but the other cited references fail to teach it as well, either singly or in combination. Accordingly, applicants respectfully submit claim claims 1-10 distinguish patentably over the references cited and request that the outstanding rejection be withdrawn.

Respectfully submitted,
E.-M. HAMANN et al.

By _____

William A. Kinnaman, Jr.
Registration No. 27,650
Phone: (845) 433-1175
Fax: (845) 432-9601

WAK/wak